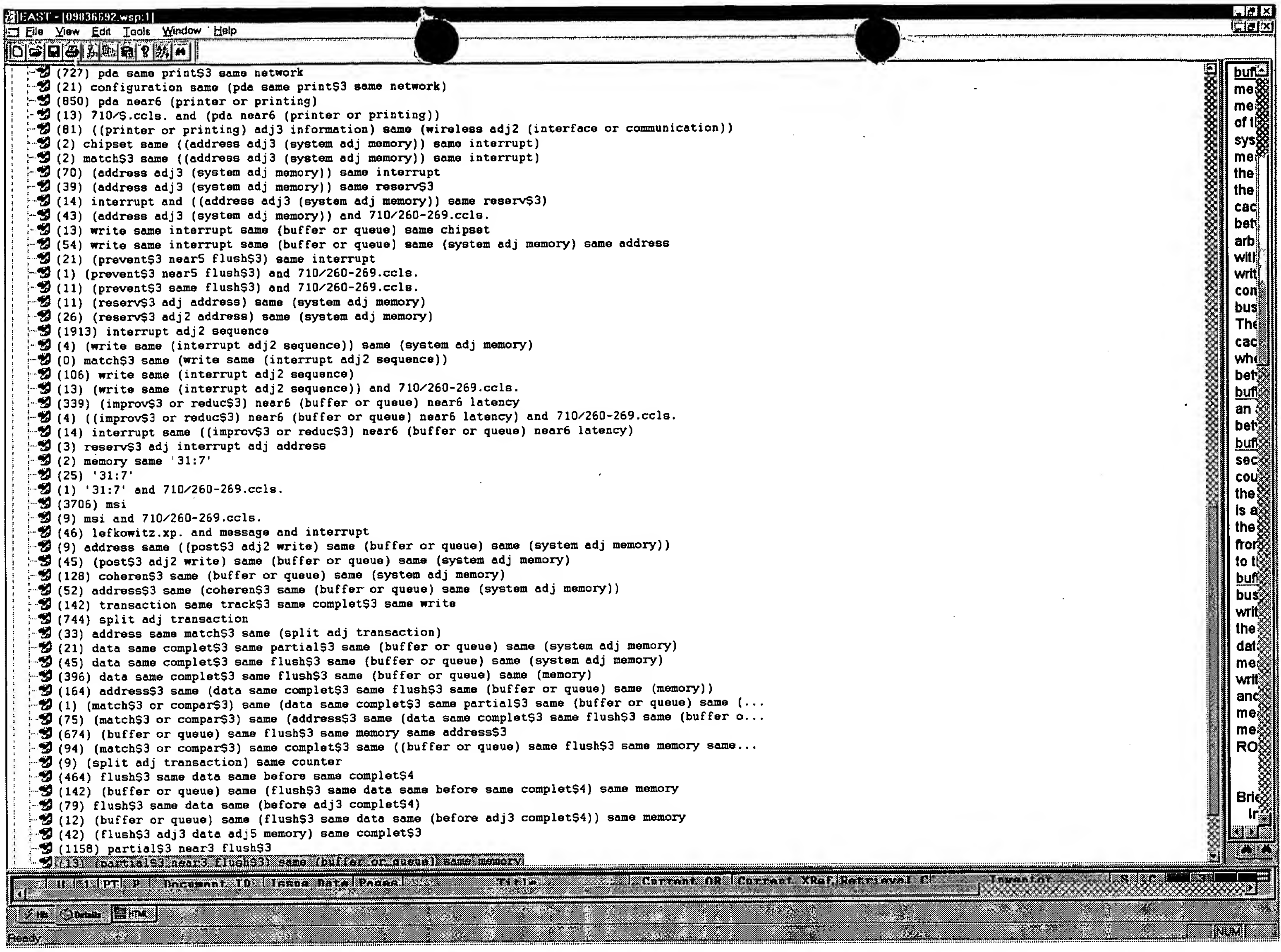
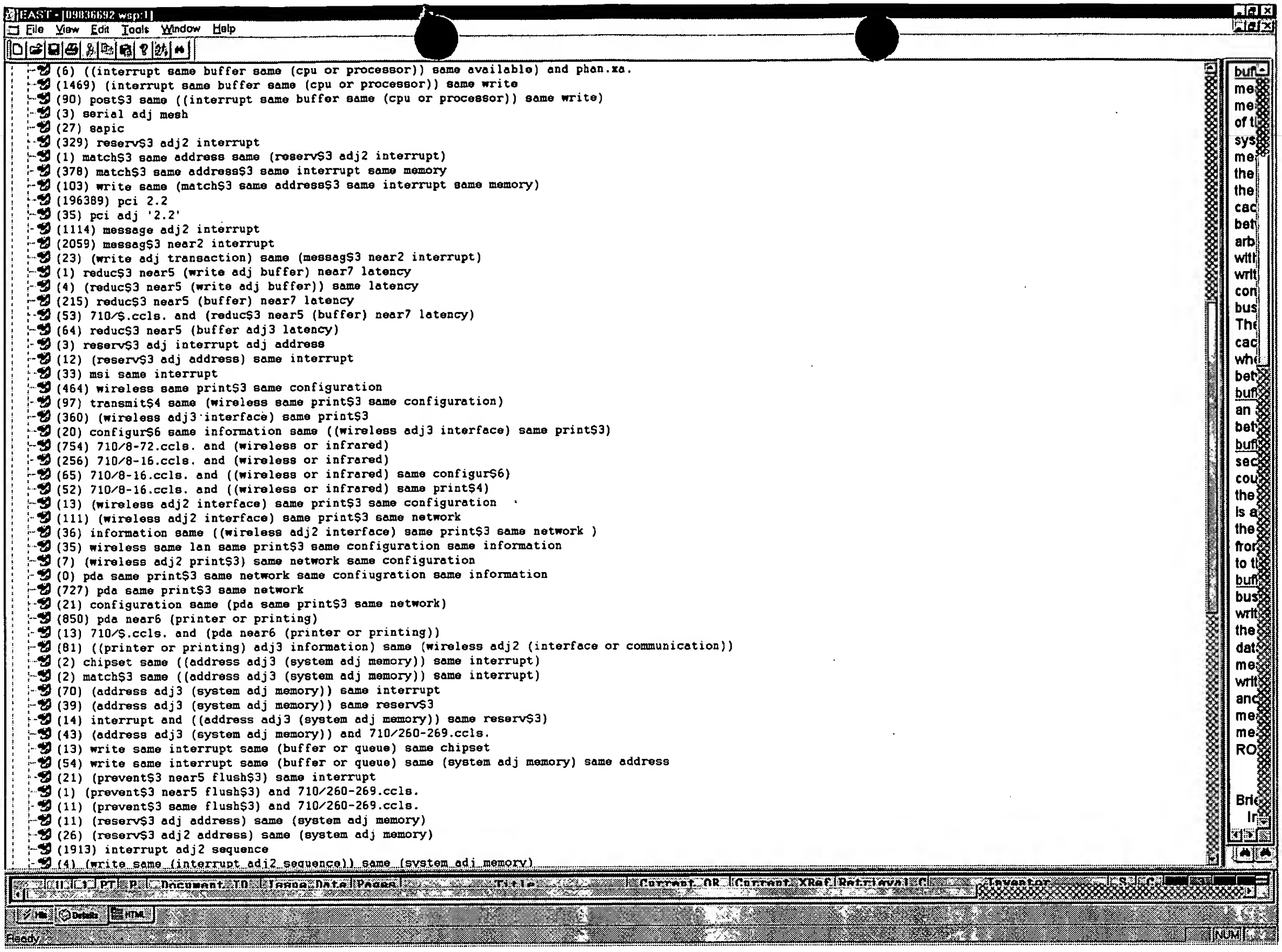


```
(727) pda same print$3 same network
(21) configuration same (pda same print$3 same network)
(850) pda near6 (printer or printing)
(13) 710/$.ccls. and (pda near6 (printer or printing))
(81) ((printer or printing) adj3 information) same (wireless adj2 (interface or communication))
(2) chipset same ((address adj3 (system adj memory)) same interrupt)
(2) match$3 same ((address adj3 (system adj memory)) same interrupt)
(70) (address adj3 (system adj memory)) same interrupt
(39) (address adj3 (system adj memory)) same reserv$3
(14) interrupt and ((address adj3 (system adj memory)) same reserv$3)
(43) (address adj3 (system adj memory)) and 710/260-269.ccls.
(13) write same interrupt same (buffer or queue) same chipset
(54) write same interrupt same (buffer or queue) same (system adj memory) same address
(21) (prevent$3 near5 flush$3) same interrupt
(1) (prevent$3 near5 flush$3) and 710/260-269.ccls.
(11) (prevent$3 same flush$3) and 710/260-269.ccls.
(11) (reserv$3 adj address) same (system adj memory)
(26) (reserv$3 adj2 address) same (system adj memory)
(1913) interrupt adj2 sequence
(4) (write same (interrupt adj2 sequence)) same (system adj memory)
(0) match$3 same (write same (interrupt adj2 sequence))
(106) write same (interrupt adj2 sequence)
(13) (write same (interrupt adj2 sequence)) and 710/260-269.ccls.
(339) (improv$3 or reduc$3) near6 (buffer or queue) near6 latency
(4) ((improv$3 or reduc$3) near6 (buffer or queue) near6 latency) and 710/260-269.ccls.
(14) interrupt same ((improv$3 or reduc$3) near6 (buffer or queue) near6 latency)
(3) reserv$3 adj interrupt adj address
(2) memory same '31:7'
(25) '31:7'
(1) '31:7' and 710/260-269.ccls.
(3706) msi
(9) msi and 710/260-269.ccls.
(46) lefkowitz.xp. and message and interrupt
(9) address same ((post$3 adj2 write) same (buffer or queue) same (system adj memory))
(45) (post$3 adj2 write) same (buffer or queue) same (system adj memory)
(128) coheren$3 same (buffer or queue) same (system adj memory)
(52) address$3 same (coheren$3 same (buffer or queue) same (system adj memory))
(142) transaction same track$3 same complet$3 same write
(744) split adj transaction
(33) address same match$3 same (split adj transaction)
(21) data same complet$3 same partial$3 same (buffer or queue) same (system adj memory)
(45) data same complet$3 same flush$3 same (buffer or queue) same (system adj memory)
(396) data same complet$3 same flush$3 same (buffer or queue) same (memory)
(164) address$3 same (data same complet$3 same flush$3 same (buffer or queue) same (memory))
(1) (match$3 or compar$3) same (data same complet$3 same partial$3 same (buffer or queue) same (...
(75) (match$3 or compar$3) same (address$3 same (data same complet$3 same flush$3 same (buffer o...
(674) (buffer or queue) same flush$3 same memory same address$3
(94) (match$3 or compar$3) same complet$3 same ((buffer or queue) same flush$3 same memory same...
(9) (split adj transaction) same counter
(464) flush$3 same data same before same complet$4
(142) (buffer or queue) same (flush$3 same data same before same complet$4) same memory
(79) flush$3 same data same (before adj3 complet$4)
(12) (buffer or queue) same (flush$3 same data same (before adj3 complet$4)) same memory
(42) (flush$3 adj3 data adj5 memory) same complet$3
(1158) partial$3 near3 flush$3
(13) (partial$3 near3 flush$3) same (buffer or queue) same memory
```



EAST - [09836692 wsp:1]
File View Edit Tools Window Help
PT P Document ID Issue Date Pages Title Current OR Current XRef Retrieval C Inventor S C
Hit Details HTML
Ready
NUM

```
(6) ((interrupt same buffer same (cpu or processor)) same available) and phan.xa.
(1469) (interrupt same buffer same (cpu or processor)) same write
(90) post$3 same ((interrupt same buffer same (cpu or processor)) same write)
(3) serial adj mesh
(27) sapic
(329) reserv$3 adj2 interrupt
(1) match$3 same address same (reserv$3 adj2 interrupt)
(378) match$3 same address$3 same interrupt same memory
(103) write same (match$3 same address$3 same interrupt same memory)
(196389) pci 2.2
(35) pci adj '2.2'
(1114) message adj2 interrupt
(2059) messag$3 near2 interrupt
(23) (write adj transaction) same (messag$3 near2 interrupt)
(1) reduc$3 near5 (write adj buffer) near7 latency
(4) (reduc$3 near5 (write adj buffer)) same latency
(215) reduc$3 near5 (buffer) near7 latency
(53) 710/$.ccls. and (reduc$3 near5 (buffer) near7 latency)
(64) reduc$3 near5 (buffer adj3 latency)
(3) reserv$3 adj interrupt adj address
(12) (reserv$3 adj address) same interrupt
(33) msi same interrupt
(464) wireless same print$3 same configuration
(97) transmit$4 same (wireless same print$3 same configuration)
(360) (wireless adj3 interface) same print$3
(20) configur$6 same information same ((wireless adj3 interface) same print$3)
(754) 710/8-72.ccls. and (wireless or infrared)
(256) 710/8-16.ccls. and (wireless or infrared)
(65) 710/8-16.ccls. and ((wireless or infrared) same configur$6)
(52) 710/8-16.ccls. and ((wireless or infrared) same print$4)
(13) (wireless adj2 interface) same print$3 same configuration
(111) (wireless adj2 interface) same print$3 same network
(36) information same ((wireless adj2 interface) same print$3 same network )
(35) wireless same lan same print$3 same configuration same information
(7) (wireless adj2 print$3) same network same configuration
(0) pda same print$3 same network same confiugration same information
(727) pda same print$3 same network
(21) configuration same (pda same print$3 same network)
(850) pda near6 (printer or printing)
(13) 710/$.ccls. and (pda near6 (printer or printing))
(81) ((printer or printing) adj3 information) same (wireless adj2 (interface or communication))
(2) chipset same ((address adj3 (system adj memory)) same interrupt)
(2) match$3 same ((address adj3 (system adj memory)) same interrupt)
(70) (address adj3 (system adj memory)) same interrupt
(39) (address adj3 (system adj memory)) same reserv$3
(14) interrupt and ((address adj3 (system adj memory)) same reserv$3)
(43) (address adj3 (system adj memory)) and 710/260-269.ccls.
(13) write same interrupt same (buffer or queue) same chipset
(54) write same interrupt same (buffer or queue) same (system adj memory) same address
(21) (prevent$3 near5 flush$3) same interrupt
(1) (prevent$3 near5 flush$3) and 710/260-269.ccls.
(11) (prevent$3 same flush$3) and 710/260-269.ccls.
(11) (reserv$3 adj address) same (system adj memory)
(26) (reserv$3 adj2 address) same (system adj memory)
(1913) interrupt adj2 sequence
(4) (write same (interrupt adj2 sequence)) same (system adj memory)
```

buf
me
me
of t
sys
me
the
the
cac
bet
arb
wit
writ
con
bus
The
cac
whe
bet
buf
an
bet
buf
sec
cou
the
is a
the
fror
to t
buf
bus
writ
the
dat
me
writ
anc
me
me
RO

Bri
Ir

US005535341A

United States Patent [19]

Shah et al.

[11] **Patent Number: 5,535,341**

[45] **Date of Patent: Jul. 9, 1996**

[54] **APPARATUS AND METHOD FOR DETERMINING THE STATUS OF DATA BUFFERS IN A BRIDGE BETWEEN TWO BUSES DURING A FLUSH OPERATION**

[75] Inventors: **Nilesh Shah; Rajeev Prasad**, both of Folsom, Calif.

[73] Assignee: **Intel Corporation**, Santa Clara, Calif.

[21] Appl. No.: **201,819**

[22] Filed: **Feb. 24, 1994**

[51] **Int. Cl.^{6}** **G06F 3/04**; G06F 13/00

[52] **U.S. Cl.** **395/306**; 395/309; 395/872; 395/878; 370/85.13

[58] **Field of Search** 395/325, 306, 395/309, 872, 879, 878; 370/85.13

[56] **References Cited**

U.S. PATENT DOCUMENTS

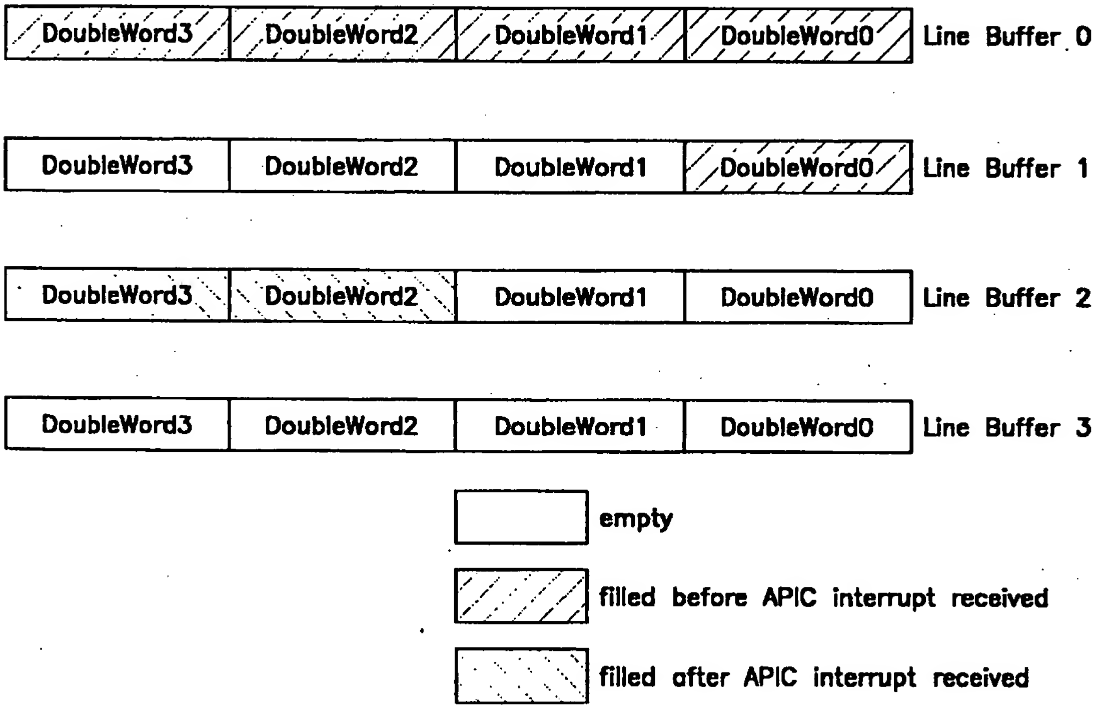
5,117,486	5/1992	Clark et al.	395/250
5,179,706	1/1993	Swanson et al.	395/725

Primary Examiner—Jack B. Harvey
Assistant Examiner—Jeffrey K. Seto
Attorney, Agent, or Firm—Blakely, Sokoloff, Taylor & Zafman

[57] **ABSTRACT**

A bridge circuit adapted to be associated with first and second bus circuits which bridge circuit includes a first path including a plurality of buffers for storing data or addresses being transferred from the second bus to the first bus, a circuit arrangement for detecting that an interrupt of a presenting-running operation has occurred, a circuit arrangement for determining the state of the plurality of buffers when an interrupt occurs, and apparatus for flushing only those buffers of the plurality which were storing data for transfer when the interrupt occurred.

23 Claims, 7 Drawing Sheets



US006629179B1

(12) **United States Patent**
Bashford

(10) **Patent No.: US 6,629,179 B1**
(45) **Date of Patent: Sep. 30, 2003**

(54) **MESSAGE SIGNALED INTERRUPT GENERATING DEVICE AND METHOD**

(75) Inventor: **Patrick R. Bashford**, Fort Collins, CO (US)

(73) Assignee: **Adaptec, Inc.**, Milpitas, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 409 days.

(21) Appl. No.: **09/630,341**

(22) Filed: **Jul. 31, 2000**

(51) **Int. Cl.**^{7} **G06F 13/24**

(52) **U.S. Cl.** **710/260**; 710/112; 710/306

(58) **Field of Search** 710/36, 306, 305, 710/260–266, 311, 314, 312, 310

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,692,200	A	*	11/1997	Carlson et al.	710/262
6,253,275	B1	*	6/2001	Waldron et al.	710/260
6,370,607	B1	*	4/2002	Williams et al.	710/262
6,389,526	B1	*	5/2002	Keller et al.	712/30

OTHER PUBLICATIONS

Unknown, "*PCI Local Bus Specification Revision 2.2*," Dec. 18, 1998, PCI Special Interest Group, 322 pages.

* cited by examiner

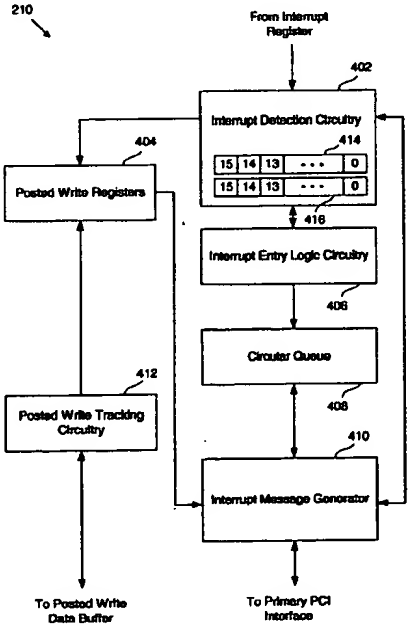
Primary Examiner—Rupal Dharia
Assistant Examiner—Kim T. Huynh

(74) *Attorney, Agent, or Firm*—Martine & Penilla, LLP

(57) **ABSTRACT**

The present invention provides a bridge device and a method for generating message signaled interrupts to indicate completion of write transactions from one or more secondary bus devices to a primary bus device. The bridge device is coupled between a first bus and a second bus. The one or more secondary bus devices are coupled to the second bus and the primary bus device is coupled to the first bus. The bridge device includes a bridge FIFO and control circuitry, a first register, and an interrupt generation logic. The bridge FIFO and control circuitry is arranged to control data transfer between the one or more secondary bus devices and the primary bus device. The bridge FIFO and control circuitry is further configured to store and transfer write data from the one or more secondary bus devices to the primary bus device. The first register is arranged to store a set of interrupt bit numbers. Each of the one or more secondary bus devices is configured to write an interrupt bit number into the first register after completion of a write data transfer to the bridge FIFO and control circuitry to indicate completion of the write data transfer. The interrupt generation logic is coupled to the bridge FIFO and control circuitry and the first register, and is arranged to generate message signaled interrupts in response to the writing of the interrupt bit numbers. In this configuration, the interrupt generation logic generates the message signaled interrupts in the order the write data transfers are posted to the first bus. In addition, each of the message signaled interrupts is generated and posted after all write data transfers associated with the interrupt bit number have been posted to the first bus.

29 Claims, 10 Drawing Sheets


210
From Interrupt Register
402
Interrupt Detection Circuitry
414
15 14 13 ... 0
15 14 13 ... 0
416
404
Posted Write Registers
Interrupt Entry Logic Circuitry
406
Circular Queue
408
410
412
Posted Write Tracking Circuitry
Interrupt Message Generator
To Posted Write Data Buffer
To Primary PCI Interface

US006298410B1

(12) **United States Patent**
Jayakumar et al.

(10) **Patent No.: US 6,298,410 B1**
(45) **Date of Patent: *Oct. 2, 2001**

(54) **APPARATUS AND METHOD FOR INITIATING HARDWARE PRIORITY MANAGEMENT BY SOFTWARE CONTROLLED REGISTER ACCESS**

(75) Inventors: **Muthurajan Jayakumar**, Sunnyvale; **Vijay Kumar Goru**; **Ravi Eakambaram**, both of San Jose, all of CA (US)

(73) Assignee: **Intel Corporation**, Santa Clara, CA (US)

(*) Notice: This patent issued on a continued prosecution application filed under 37 CFR 1.53(d), and is subject to the twenty year patent term provisions of 35 U.S.C. 154(a)(2).

Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/001,817**

(22) Filed: **Dec. 31, 1997**

(51) **Int. Cl.**[7] **G06F 13/24**; G06F 13/26; G06F 13/34

(52) **U.S. Cl.** **710/266**; 710/264; 710/267; 710/261; 710/262; 712/32

(58) **Field of Search** 710/113, 261, 710/262, 263, 264, 266, 267, 269, 265, 130, 217; 708/672; 712/32; 713/502

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,365,294	*	12/1982	Stokken	713/502
4,604,683		8/1986	Russ et al. .	
5,384,724	*	1/1995	Jagini	708/672
5,446,910		8/1995	Kennedy et al. .	
5,471,620	*	11/1995	Shimizu et al.	710/265
5,481,728	*	1/1996	Matsutani	710/113
5,491,828		2/1996	Intrater et al. .	

OTHER PUBLICATIONS

PCT/US98/27514, PCT International Search Report.

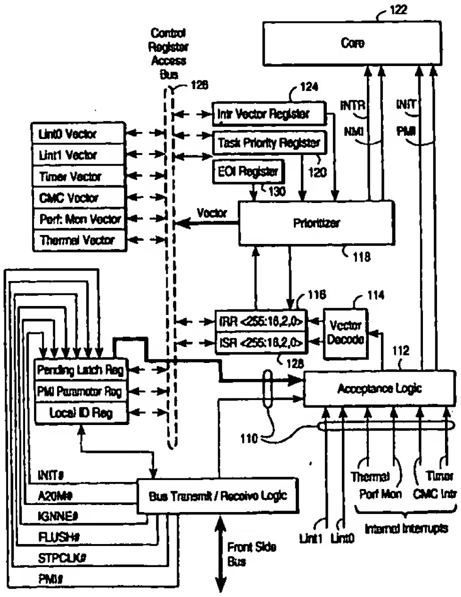
* cited by examiner

Primary Examiner—Daniel H. Pan
(74) *Attorney, Agent, or Firm*—Blakely, Sokoloff, Taylor & Zafamn LLP

(57) **ABSTRACT**

An apparatus and method for controlling interrupts in a computer are disclosed, in which programmable software operates to control when data concerning the interrupt having highest priority is to be provided, and hardware logic operates to control how that data is provided. An interrupt vector register is included in the computer CPU. The interrupt vector register does not act like the typical register. It is not a physical register, and cannot be written to. A read to this register by the programmable software, triggers the hardware logic. Once triggered, this logic performs certain control tasks, the end result of which is returning to the programmable software, a vector corresponding to the interrupt having highest priority. The programmable software can implement various software policies, in addition to the hardware policy implemented by the hardware logic.

22 Claims, 4 Drawing Sheets


122
Core
Control Register Access Bus
126
124
Intr Vector Register
INTR
INIT
NMI
PMI
Lint0 Vector
Lint1 Vector
Timer Vector
CMC Vector
Perf: Mon Vector
Thermal Vector
Task Priority Register
EOI Register
120
130
Vector
Prioritizer
118
116
114
IRR <255:16,2,0>
ISR <255:16,2,0>
Vector Decode
112
128
Pending Latch Reg
PMI Parameter Reg
Local ID Reg
Acceptance Logic
110
INIT#
A20M#
IGNNE#
FLUSH#
STPCLK#
PMI#
Bus Transmit / Receive Logic
Front Side Bus
Thermal
Perf Mon
Timer
CMC Intr
Lint1
Lint0
Internal Interrupts